

Gradient based aerodynamic shape optimization using the FIVER embedded boundary method

D. De Santis*, M. J. Zahr† and C. Farhat‡

Stanford University, Stanford, CA 94305, USA

We present an approach for the computation of the shape sensitivities within the embedded boundary method for compressible flows. The derivatives are used in a gradient-based optimization procedure to perform aerodynamic shape optimization of complex configurations. The fluid solver uses the finite volume solver with the FIVER approach recently developed to perform accurate and robust numerical simulations of complex fluid-structure interaction problems. An analytical derivation of the sensitivities is reported and the performance of the optimization algorithm is demonstrated with numerical tests.

I. Introduction

The continuous advance in numerical methods and the large availability of computational power have made Computational Fluid dynamics (CFD) a reliable tool for the design and the development in many engendering fields, and the aeronautical one in particular. One advantage of the CFD, over the experimental approach, is that numerical simulations can be used not only to study the performance of a single configuration but they give the flexibility to study the effect that changes in the baseline configuration have on the performance. When combined with optimization procedures, CFD has the potential to delivery the best and most robust product.

Among several different optimization methods, gradient-based optimization is an effective approach for aerodynamic design problems. In gradient-based optimization methods, having accurate derivatives of the objective function is crucial in order to ensure robust and efficient convergence of the optimization algorithm, in other words the effectiveness of the optimization is directly related to the accuracy of the gradients. Adjoint and direct approaches can be used to compute accurate sensitivities and to reduce the evaluation cost compared to a crude approach such as finite differences.

For aerodynamic shape optimization problems, the design variables usually involves parameters that modify the shape of a given initial geometry. For a certain objective function, the design variables are used to define a deformed geometry at each optimization cycle, which is used to compute the new flow field, required to evaluate the objective function. In the traditional body-fitted approach, on structured or unstructured meshes, a change in the shape of the surface mesh requires to propagate smoothly the deformation also in the volume grid to avoid large distortions and interpenetration of neighboring elements in the CFD mesh. Various mesh deformation methods (*e.g.* spring stiffness,² elastic analogy) have been proposed to tackle this problem, however as the geometric complexity of optimization problems increases, the robustness of this approaches reduces and the computational cost of the mesh deformation becomes not negligible. In addition, the linearization of the mesh-perturbation scheme, (*i.e.* mesh sensitivity), must be computed to take into account the effect of shape perturbations in the flow equations, with an addition cost and complexity.

Embedded boundary methods provide a promising alternative to the classical body-fitted discretizations. In the embedded boundary approach, the CFD grid does not conform to the geometry of the object, eliminating the problem of the mesh deformation. When combined with an efficient flow solver this approach

*Postdoctoral Research Fellow. Email: ddesanti@stanford.edu

†Graduate Student. Email: mzahr@stanford.edu

‡Vivian Church Hoff Professor of Aircraft Structures; AIAA Fellow. Email: cfarhat@stanford.edu

is particularly well suited for the automated analysis of complex geometry problems. An additional advantage of non boundary-conforming numerical methods is that there is no need to compute the internal mesh sensitivity with a drastically reduction of user coding and computation effort.

However, operating on nonbody-fitted grids has its own difficulties, especially in the imposition of the wall boundary condition. In particular, many embedded boundary methods tend to be only first-order accurate in space at the embedded fluid–structure interfaces. For compressible inviscid and viscous flows, a new approach for the treatment of a fluid–structure interface was proposed in^{6,13} in the contest of finite volume (FV) methods. This approach is based on ideas first presented in⁴ for the treatment of fluid–fluid interfaces in multi-phase flow problems. It differs from most embedded boundary methods for the fact that instead of relying primarily on interpolation and/or extrapolation, the method enforces the appropriate value of the fluid velocity at the fluid–structure interface and recovers the value of the fluid pressure and density at this location by solving local, one-dimensional, exact fluid-structure half Riemann problem. For this reason this approach has been called FIVER: Finite Volume method with exact Riemann solvers.

Our goal is to combine the flexibility of the FIVER based embedded boundary approach with an efficient computation of design sensitivities, in order to produce a powerful combination CFD–optimization able to run fast, accurate and automatic shape optimization problems.

II. Optimization problem formulation

In this work, the discretize-then-differentiate approach to PDE-constrained optimization is used, whereby the spatial discretization is applied to the *steady* PDE and the optimality system of the discrete problem is derived. In the context of optimization, this approach has the advantage of ensuring the gradients of output functionals will be consistent with the functionals themselves, since the discretization errors are differentiated. The fully-discrete optimization problem takes the form

$$\begin{aligned} & \underset{\mathbf{w} \in \mathbb{R}^{N_{\mathbf{w}}}, \boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} && \mathcal{J}(\mathbf{w}, \boldsymbol{\mu}) \\ & \text{subject to} && \mathbf{R}(\mathbf{w}, \boldsymbol{\mu}) = 0 \\ & && \mathbf{c}(\mathbf{w}, \boldsymbol{\mu}) \leq 0 \end{aligned} \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^{N_{\mathbf{w}}}$ is the discretized PDE state vector, $\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}$ are the shape parameters, $\mathcal{J} : \mathbb{R}^{N_{\mathbf{w}}} \times \mathbb{R}^{N_{\boldsymbol{\mu}}} \rightarrow \mathbb{R}$ is the discretized objective function, $\mathbf{c} : \mathbb{R}^{N_{\mathbf{w}}} \times \mathbb{R}^{N_{\boldsymbol{\mu}}} \rightarrow \mathbb{R}^{N_{\mathbf{c}}}$ are the $N_{\mathbf{c}}$ additional constraints (discretized), and $\mathbf{R}(\mathbf{w}, \boldsymbol{\mu}) = 0$ is the discretized PDE.

The nested approach to PDE-constrained optimization will be considered, where the state vector is viewed as an implicit function of the parameters, $\mathbf{w} = \mathbf{w}(\boldsymbol{\mu})$, through application of the implicit function theorem to the discretized PDE, $\mathbf{R}(\mathbf{w}, \boldsymbol{\mu}) = 0$. Utilizing this implicit relationship between the state vector and parameters, the optimization problem in (1) reduces to

$$\begin{aligned} & \underset{\boldsymbol{\mu} \in \mathbb{R}^{N_{\boldsymbol{\mu}}}}{\text{minimize}} && \mathcal{J}(\mathbf{w}(\boldsymbol{\mu}), \boldsymbol{\mu}) \\ & \text{subject to} && \mathbf{c}(\mathbf{w}(\boldsymbol{\mu}), \boldsymbol{\mu}) \leq 0. \end{aligned} \tag{2}$$

Application of *gradient-based* optimization techniques to solve (2) requires gradients of both the objective and constraints, i.e. $\frac{d\mathcal{J}}{d\boldsymbol{\mu}}$ and $\frac{d\mathbf{c}}{d\boldsymbol{\mu}}$. In the remainder of this section, $\mathcal{J}(\mathbf{w}, \boldsymbol{\mu})$ will be used to denote any optimization functional whose gradient is desired, i.e. either the objective function or a constraint.

Derivation of the gradient of the optimization functional, $\mathcal{J}(\mathbf{w}(\boldsymbol{\mu}), \boldsymbol{\mu})$, proceeds by expanding $\frac{d\mathcal{J}}{d\boldsymbol{\mu}}$ via the chain rule

$$\frac{d\mathcal{J}}{d\boldsymbol{\mu}} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{\mu}} + \frac{\partial \mathcal{J}}{\partial \mathbf{w}} \frac{\partial \mathbf{w}}{\partial \boldsymbol{\mu}}. \tag{3}$$

The partial derivatives $\frac{\partial \mathcal{J}}{\partial \boldsymbol{\mu}}$ and $\frac{\partial \mathcal{J}}{\partial \mathbf{w}}$ are analytically computable from specific form of $\mathcal{J}(\mathbf{w}, \boldsymbol{\mu})$. An expression for the state vector sensitivity, $\frac{\partial \mathbf{w}}{\partial \boldsymbol{\mu}}$ is obtained by differentiating

$$\mathbf{R}(\mathbf{w}(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0, \tag{4}$$

which must be satisfied for all $\boldsymbol{\mu} \in \mathbb{R}^{N_\mu}$, given that \mathbf{w} is an implicit function of $\boldsymbol{\mu}$ through this exact equation, $\mathbf{R}(\mathbf{w}, \boldsymbol{\mu}) = 0$. Differentiation of (4) leads to the expression

$$\frac{d\mathbf{R}}{d\boldsymbol{\mu}} = \frac{\partial\mathbf{R}}{\partial\boldsymbol{\mu}} + \frac{\partial\mathbf{R}}{\partial\mathbf{w}} \frac{\partial\mathbf{w}}{\partial\boldsymbol{\mu}} = 0. \quad (5)$$

From (5), it is clear the expression for the state sensitivity, $\frac{d\mathbf{w}}{d\boldsymbol{\mu}}$, is

$$\frac{\partial\mathbf{w}}{\partial\boldsymbol{\mu}} = - \left[\frac{\partial\mathbf{R}}{\partial\mathbf{w}} \right]^{-1} \frac{\partial\mathbf{R}}{\partial\boldsymbol{\mu}} \quad (6)$$

To complete the expression for the gradient of the functional, \mathcal{J} , the expression for the sensitivity (6) is substituted into (3)

$$\frac{d\mathcal{J}}{d\boldsymbol{\mu}} = \frac{\partial\mathcal{J}}{\partial\boldsymbol{\mu}} - \frac{\partial\mathcal{J}}{\partial\mathbf{w}} \left(\left[\frac{\partial\mathbf{R}}{\partial\mathbf{w}} \right]^{-1} \frac{\partial\mathbf{R}}{\partial\boldsymbol{\mu}} \right), \quad (7)$$

which is known as the sensitivity, or direct, approach for computing $\frac{d\mathcal{J}}{d\boldsymbol{\mu}}$. Slightly re-arranging this expression leads to adjoint method

$$\frac{d\mathcal{J}}{d\boldsymbol{\mu}} = \frac{\partial\mathcal{J}}{\partial\boldsymbol{\mu}} - \left(\left[\frac{\partial\mathbf{R}}{\partial\mathbf{w}} \right]^{-T} \frac{\partial\mathcal{J}}{\partial\mathbf{w}} \right)^T \frac{\partial\mathbf{R}}{\partial\boldsymbol{\mu}}. \quad (8)$$

It is clear that computing $\frac{d\mathcal{J}}{d\boldsymbol{\mu}}$ via the sensitivity approach requires the solution N_μ linear systems of equations (of size N_w) and a single matrix-vector product for each functional \mathcal{J} . In contrast, the adjoint method requires a single linear solve (of size N_w) for each functional \mathcal{J} and N_w matrix-vector products. Regardless of whether the sensitivity or adjoint method are employed to compute functional gradients, the following terms are required: $\frac{\partial\mathcal{J}}{\partial\mathbf{w}}$, $\frac{\partial\mathcal{J}}{\partial\boldsymbol{\mu}}$, $\frac{\partial\mathbf{R}}{\partial\mathbf{w}}$, $\frac{\partial\mathbf{R}}{\partial\boldsymbol{\mu}}$. Additionally, these are *exactly* the same terms that would be required if gradient-based optimization was applied to (1) directly, i.e. the one-shot approach to PDE-constrained optimization where the optimal parameter configuration and PDE solution are computed *simultaneously*.

The remainder of this document will be devoted to the derivation of the necessary terms, $\frac{\partial\mathcal{J}}{\partial\mathbf{w}}$, $\frac{\partial\mathcal{J}}{\partial\boldsymbol{\mu}}$, $\frac{\partial\mathbf{R}}{\partial\mathbf{w}}$, $\frac{\partial\mathbf{R}}{\partial\boldsymbol{\mu}}$, for the FIVER embedded boundary method, as well as demonstration of the benefits of using an embedded boundary method for aerodynamic shape optimization.

III. Flow solver and analytical sensitivities in the embedded framework

A. The FIVER solver

FIVER is a FV method based on the solution of exact, one-dimensional, Riemann problems for the simulation of multi-material flows on Eulerian meshes. It was first introduced in^{3,4} in the context of two-phase compressible flow problems characterized by large density jumps. Next, it was extended to FSI problem for both inviscid and viscous flows.^{5,6,12,13}

The governing equations for an inviscid fluid are semi-discretized in space by a vertex-based FV method on unstructured grid. The case of a structured mesh is treated as a particular case. The main goal of the FIVER approach is to keep the semi-discretization of the governing equations as close as possible to the classical FV scheme. To this purpose, let us consider the set of non-overlapping control volumes \mathcal{C}_i , with boundary $\partial\mathcal{C}_i$, surrounding each node i of the CFD grid. It is well known that in the FV schemes the residual on the control volume \mathcal{C}_i is expressed as

$$\mathbf{R}_i = \sum_{j \in \mathcal{K}_i} \mathbf{F}_{ij}(\mathbf{w}_{ij}, \mathbf{w}_{ji}, \mathbf{n}_{ij}), \quad (9)$$

where \mathcal{K}_i is the set of the vertices connected by an edge to vertex i and \mathbf{F}_{ij} is the numerical flux function at the cell interface $\partial\mathcal{C}_{ij} = \partial\mathcal{C}_i \cap \partial\mathcal{C}_j$ with outward normal vector \mathbf{n}_{ij} . Following the MUSCL approach,¹¹

the states \mathbf{w}_{ij} , \mathbf{w}_{ji} in Eq. (9) are the extrapolated and limited fluid states from the i -side and j -side, respectively. This ensures second order spatial accuracy away from shocks and monotonicity of the solution across discontinuities.

When a control volume \mathcal{C}_i is traversed by the material interface, the spatial approximation must account for the presence of the structure. To this purpose, FIVER relies on tracking the fluid–structure interface using a robust intersector,¹³ capable of handling open thin shell and closed surfaces, even in the case of an under-resolved CFD grid. The intersector is used to identify the edges that intersect the structure and determine if a node is in the fluid domain or is covered by the embedded structure.

In the simplest version of FIVER, the fluid–structure interface is approximated by a surrogate counterpart which coincides with the control volume interface, Fig. 1. Supposing, for example, that the node i belongs to the fluid domain and the node j is covered by the embedded structure, in the FIVER approach the numerical flux \mathbf{F}_{ij} is computed as follows

$$\mathbf{F}_{ij} = \mathbf{F}_{ij}(\mathbf{w}_{ij}, \mathbf{w}_{ij}^*, \mathbf{n}_{ij}),$$

where \mathbf{w}_{ij}^* denotes the exact solution of the one-dimensional, fluid–structure half Riemann problem at the surrogate interface, based on the known normal component of the velocity of the structure and the fluid state at the node i . This approach introduces a spatial error of order $\mathcal{O}(h/2)$ in the vicinity of the discrete embedded surface, with h the grid size of the CFD grid.

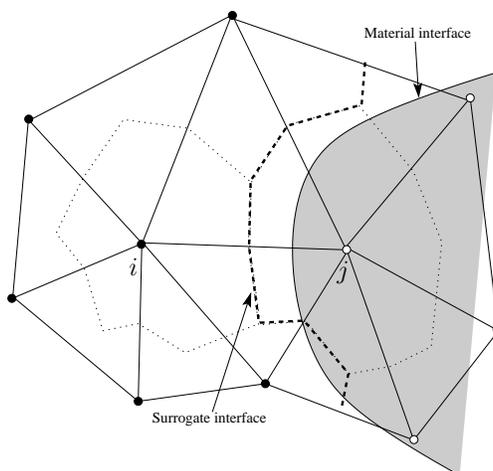


Figure 1. Definition of the surrogate and material interfaces.

The extension of the FIVER approach to the second order accurate spatial discretization near the embedded surface is discussed in¹⁴ for structured grids and in⁸ for the general case of three-dimensional unstructured grids. Here the approach is only briefly recalled. The main idea is to construct and solve the one-dimensional Riemann problems at the material interface (where the discrete embedded surface is actually located) instead of the surrogate interface. This objective is achieved with the following steps:

1. The fluid state is extrapolated to the material interface Γ using the fluid states of the nodes in the fluid domain and the corresponding nodal gradients:

$$\mathbf{w}_\Gamma = \mathbf{w}_i + \nabla \mathbf{w}_i \cdot (\mathbf{x}_\Gamma - \mathbf{x}_i), \quad (10)$$

where \mathbf{x}_Γ is the location of the intersection between edge i – j and the discrete embedded surface, \mathbf{x}_i is the location of node i , and $\nabla \mathbf{w}_i$ is the nodal gradient of the solution at node i .

2. The one-dimensional half-Riemann problem is solved exactly at the material interface using the state \mathbf{w}_Γ and the normal component of the velocity of the structure at \mathbf{x}_Γ as input values. The solution of this half-Riemann problem is indicated by \mathbf{w}_i^* .
3. The fluid state is interpolated/extrapolated at control volume interface $\partial\mathcal{C}_{ij}$ as follows:

$$\mathbf{w}_{ij}^* = \frac{\|\mathbf{x}_{ij} - \mathbf{x}'_i\| \mathbf{w}_i^* - \|\mathbf{x}_{ij} - \mathbf{x}_\Gamma\| \mathbf{w}'_i}{\|\mathbf{x}_{ij} - \mathbf{x}'_i\|}, \quad (11)$$

where \mathbf{x}_{ij} is the location of the control volume interface on the edge $i-j$ and, with reference to Fig. 2, \mathbf{x}'_i is the position of the point located at the intersection of the extension of edge $i-j$ and the edge of the triangle Δii_1i_2 opposite to the node i . Note that, although for simplicity the description here is restricted to two dimensional grids of triangles, the same approach is valid for three dimensional grids with hybrid elements.

4. Finally, the flux at the control volume interface is computed as $\mathbf{F}_{ij} = \mathbf{f}(\mathbf{w}_{ij}^*)$, with $\mathbf{f} = \mathbf{f}(\mathbf{w})$ the flux function of the Euler equations.

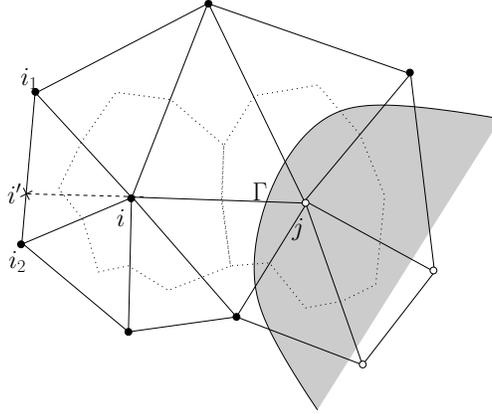


Figure 2. Schematic illustration of the construction of the second order FIVER method for unstructured grids.

The calculation of the aerodynamic forces and the flow-induced loads within the embedded framework are fraught with technical challenges due to the lack of direct availability of the computed CFD results on the embedded discrete surface. In¹³ two different approaches for the computation of flow-induced loads on a surrogate discrete surface were presented. The first approach computes the flow-induced load on the surrogate interface, which is a crude approximation of the real surface and can lead to inaccurate force calculation in the case of viscous flows. The second approach computes the aerodynamic loads on an embedded surface locally reconstructed from the knowledge of the points of intersection of the edges of the CFD-grid with the discrete embedded surface. This approach guaranties accurate calculations of the fluid-induced load in both viscous and inviscid cases, but unfortunately the process of the local reconstruction of the embedded surface suffers from robustness issue.

In this work we adopt the approach described in⁶ for the calculation of the aerodynamic forces. Unlike the other two approaches discussed earlier, this method operates directly on the discrete embedded surface and therefore it avoids the problems associated with the reconstruction of the surrogate embedded discrete surface. The starting point of the load computation method is the finite element expression of the generalized force vector \mathbf{f} associated with the pressure field (and possibly the viscous stress tensor). For each node i of the each element e of the discrete embedded surface, the discrete expression of the force vector is

$$\mathbf{f}_i = \sum_{q=1}^{N_q} \omega_q \psi_i^e(\mathbf{x}_q) (-P(\mathbf{x}_q)) \mathbf{n}^e \Delta s, \quad (12)$$

where ψ_i^e is the finite element basis function on the element e associated with the node i , the pair (ω_q, \mathbf{x}_q) is the pair of weight and coordinates of Gauss quadrature point, and Δs is the area of the element e with unit normal \mathbf{n}^e . The value of the pressure at the quadrature point, $P(\mathbf{x}_q)$, is obtained by linear extrapolation of the solution from the nearest node in the fluid domain.

B. Computation of the analytical sensitivities

At the heart of the gradient-based optimization is an accurate evaluation of the gradient of objective function, which in the case of the flow-sensitivity approach is computed as follows (see Eq. (7)):

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{w}} \right] \frac{d\mathbf{w}}{d\boldsymbol{\mu}} = - \frac{\partial \mathbf{R}}{\partial \boldsymbol{\mu}} \quad (13)$$

$$\frac{d\mathcal{J}}{d\boldsymbol{\mu}} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{\mu}} + \frac{\partial \mathcal{J}}{\partial \mathbf{w}} \frac{d\mathbf{w}}{d\boldsymbol{\mu}} \quad (14)$$

The first step requires the solution of the linearized flow problem given by Eq. (13), in which the term $\left[\frac{\partial \mathbf{R}}{\partial \mathbf{w}} \right]$ represents the Jacobian of the discretized flow equations. For internal nodes far from the discrete embedded surface the Jacobian involves the linearization of the numerical flux, MUSCL reconstruction and limiting of the solution, as well as the linearization of the far-field boundary conditions. Details about the analytical calculation of these entries in the Jacobian matrix can be found in.⁷

The contribution to the Jacobian due to the embedded boundary treatment can be easily computed by chain rules as

$$\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{w}} = \frac{\partial \mathbf{f}(\mathbf{w}_{ij}^*)}{\partial \mathbf{w}_{ij}^*} \frac{\partial \mathbf{w}_{ij}^*}{\partial \mathbf{w}}.$$

With Eq. (11) in mind, it is easy to see that the term $\partial \mathbf{w}_{ij}^*/\partial \mathbf{w}$ can be computed as follows:

$$\frac{\partial \mathbf{w}_i^*}{\partial \mathbf{w}} = \frac{\partial \mathbf{w}_i^*}{\partial \mathbf{w}_\Gamma} \frac{\partial \mathbf{w}_\Gamma}{\partial \mathbf{w}},$$

where the first term on the right-hand side of the previous equation is the the derivative of the solution of the half-Riemann problem with respect to the fluid state variable,⁹ while the term $\partial \mathbf{w}_\Gamma/\partial \mathbf{w}$ is obtained by differentiating Eq. (10).

The term $\partial \mathbf{R}/\partial \boldsymbol{\mu}$ in Eq. (13), for the embedded boundary approach, is non zero only for the nodes close to the material interface and represents the direct effect of the deformation of the geometry on the numerical residual. Once a parametrization $\mathbf{x}_\Gamma = \mathbf{x}_\Gamma(\boldsymbol{\mu})$ of the discrete embedded surface is introduced (with known shape sensitivity $d\mathbf{x}_\Gamma/d\boldsymbol{\mu}$) we can easily write

$$\frac{\partial \mathbf{R}_i}{\partial \boldsymbol{\mu}} = \frac{\partial \mathbf{f}(\mathbf{w}_{ij}^*)}{\partial \mathbf{w}_{ij}^*} \frac{\partial \mathbf{w}_{ij}^*}{\partial \boldsymbol{\mu}},$$

where the last term on the right-hand side of the previous equation is computed by taking the derivative with the respect to shape design parameter of the geometrical quantities in Eq. (10) and Eq. (11).

Because the objective functions considered in this work, are based on lift and drag coefficients, the derivative of the objective functions ultimately requires to compute the derivative of Eq. (12). Due to the fact that the flow-induced forces are computed directly on the discrete embedded surface, the derivatives with respect to the shape design variables, $d\mathbf{x}_q/d\boldsymbol{\mu}$ and $d(\mathbf{n}_e \Delta s)/d\boldsymbol{\mu}$, can be easily computed. In addition, due to the linear extrapolation of the pressure to the discrete embedded surface, the derivative $\partial \mathcal{J}/\partial \mathbf{w}$ in Eq. (14) requires the calculation of the derivative of an expression similar to that reported in Eq. (10).

An important feature of the embedded boundary approach is that the mesh metric quantities of the interior elements in the CFD grid are decoupled from the deformation of the embedded surface. For boundary-conforming grids, the modification of the surface requires the deformation also of the internal computational mesh. Although, different approaches have been proposed to propagate smoothly the surface displacements into the domain and maintain good mesh quality, the large surface deformations involved in shape optimization problems can produce very distorted grids and even negative mesh cell volumes. In addition, for body-fitted grids, the complete evaluation of the objective function sensitivities requires the evaluation of the mesh sensitivities of both the surface and the interior elements, which substantially complicate the evaluation of the term $\frac{\partial \mathbf{R}}{\partial \boldsymbol{\mu}}$ in Eq. (13).

IV. Results and discussion

We present two design examples to illustrate the effectiveness of the proposed approach. Although the examples are actually two dimensional problems, they are treated as three dimensional ones and in all the

simulations three dimensional unstructured grids of tetrahedra are used. The fluid flow is modeled using the Euler equations, which are semi-discretized in space using the approach described in the previous section. Convergence to the steady-state is obtained using a pseudo-time integration based on the backward Euler scheme. The non-linear system is solved using the Newton method and the resulting linear system is solved with a preconditioned GMRES method.

We used finite difference to verify the correctness of the analytical sensitivities, with an excellent agreement between the two approaches.

A. Single-element airfoil shape optimization

The first design example is the optimization of the NACA-0012 airfoil. The goal of the optimization is to maximize lift. The free-stream Mach number is 0.5 and the angle of attack is 2° . The computational fluid mesh contains approximately 229 000 nodes and 918 000 elements.

The SDESIGN software,¹⁰ based on the design element approach, is used for shape parametrization. The initial shape is mapped onto a set of design elements with nodal displacement degrees of freedom which are used to morph the given shape into a different one by applying to it a displacement field represented by shape functions generated by the design elements and their nodal degrees of freedom. Here, a single cubic design element is used to parametrize the NACA-0012 airfoil using this approach. Such a design element has 8 control nodes. They are used to define cubic Lagrangian polynomials for describing the displacement fields along the horizontal edges of the element, and linear functions for representing the displacement fields along its vertical ones. For this application, the set of admissible shapes generated by this representation is restricted by constraining the control nodes to move in the vertical direction only. This results in a parameterization with 8 variables where each of them represents the displacement of a control node in the vertical direction. To eliminate the case of rigid translation of the element, one of the displacement variables is constrained to zero.

In Fig. 3 the convergence of the lift is reported. The optimization process converges in approximately 10 design iterations. In Fig. 4 are reported the shape of the original NACA-0012 airfoil and the shape of

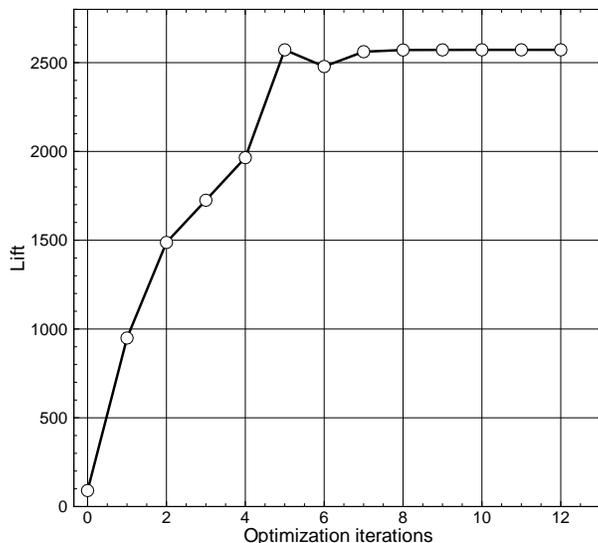


Figure 3. Convergence of the lift value for the optimization problem of the NACA-0012 airfoil

the airfoil which maximizes the lift. Note that we did not impose any constrain on the thickness of the final airfoil in order to achieve the most severe deformation of the geometry. Although in real life applications, constrains in the thickness of the airfoil must be taken into account, our objective here is to check the robustness and the flexibility of the optimization procedure within the embedded boundary approach.

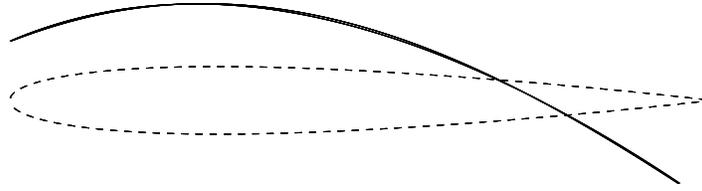


Figure 4. Original NACA-0012 airfoil (dashed line) and the final shape of the airfoil (solid line) which maximizes the lift.

B. Multi-element airfoil with large kinematics

In the second example we consider the preliminary design of a multi-element high-lift configuration. It is a three element airfoil consisting of a main element, a slat forward of the main element and a flap aft of the main element. The optimal design of high-lift devices is very challenging due to complex flow fields involved and the large number of design variables, which include not only moderate perturbations of the baseline geometry but also large displacements and deflections of the slat and flap elements.

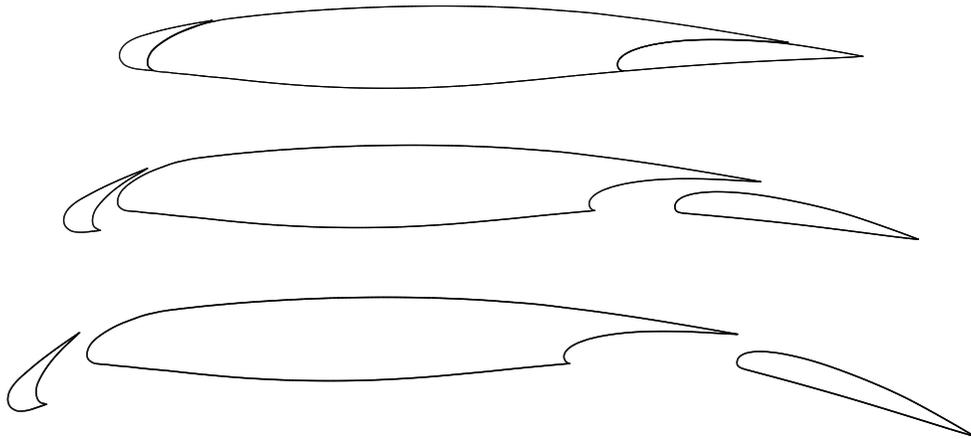


Figure 5. Multi element airfoil: deployment of the slat and flap starting from the fully closed configuration up to a wide open one.

The goal of this optimization is the maximize the lift by finding the best position of the slat and flap elements without introducing any shape modification. To exploit the capabilities of the embedded boundary approach, we start from completely closed configuration in which slat, main airfoil and flap actually form a single element airfoil (Fig. 5), we then let the optimizer find the best relative positions of the airfoil elements. This problem involves extremely large kinematics which result in topology change of the airfoil configuration. Such large displacements are unlikely to produce valid CFD grids in the case of body fitted schemes.

Six design variables are used, these are the rotation, the vertical and horizontal displacements of the slat and flap elements. A pinball algorithm¹ is use to detect any contact between the surface of the main airfoil and those of the flap/slat in order to avoid interpenetration between the airfoil elements during the optimization cycle. The same fluid grid used to perform the previous test case is used also here. The free-stream Mach number is 0.2 and the angle of attach is 10° .

Starting from the fully closed configuration, the final configuration that maximize the lift is shown Fig. 6. The final value of the lift is almost double after 6 optimization iterations.

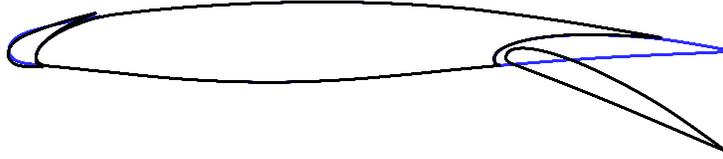


Figure 6. Optimization of the multi element airfoil to maximize the lift. Initial configuration in blue line and final configuration in black line.

V. Conclusion

We have presented a fluid sensitivity approach for computing aerodynamic shape sensitivities using three-dimensional Euler equations. The equations were discretized using an embedded boundary method based on the use of a second FV method with exact Riemann solvers. The work focused on the linearization of the discrete residual with respect to the shape design variable. The exactness of the linearization was tested with finite difference approximations. Two preliminary design problems were considered with the objective to show the potential of the embedded boundary based optimization procedure in handling large displacements and deformations of the initial configuration. Future work will focus on the extension of this approach to viscous problems and complex configurations.

References

- ¹T. Belytschko and M. O. Neal. Contact-impact by the pinball algorithm with penalty and lagrangian methods. *International Journal for Numerical Methods in Engineering*, 31(3):547–572, 1991.
- ²C. Degand and C. Farhat. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Computers & Structures*, 80(34):305 – 316, 2002.
- ³C. Farhat, J.F. Gerbeau, and A. Rallu. Fiver: A finite volume method based on exact two-phase riemann problems and sparse grids for multi-material flows with large density jumps. *Journal of Computational Physics*, 231(19):6360 – 6379, 2012.
- ⁴C. Farhat, A. Rallu, and S. Shankaran. A higher-order generalized ghost fluid method for the poor for the three-dimensional two-phase flow computation of underwater implsions. *Journal of Computational Physics*, 227(16):7674 – 7700, 2008.
- ⁵C. Farhat and Lakshminarayan V. An ALE formulation of embedded boundary methods for tracking boundary layers in turbulent fluidstructure interaction problems. *Journal of Computational Physics*, 263:53 – 70, 2014.
- ⁶V. Lakshminarayan, C. Farhat, and A. Main. An embedded boundary framework for compressible turbulent flow and fluidstructure computations on structured and unstructured grids. *International Journal for Numerical Methods in Fluids*, 76(6):366–395, 2014.
- ⁷M. Lesoinne, M. Sarkis, U. Hetmaniuk, and C. Farhat. A linearized method for the frequency analysis of three-dimensional fluid/structure interaction problems in all flow regimes. *Computer Methods in Applied Mechanics and Engineering*, 190(2425):3121 – 3146, 2001.
- ⁸A. Main. *Implicit and higher-order discretization methods for compressible multi-phase fluid and fluid-structure problems*. PhD thesis, Stanford University, 2014.
- ⁹A. Main and C. Farhat. A second-order time-accurate implicit finite volume method with exact two-phase riemann problems for compressible multi-phase fluid and fluidstructure problems. *Journal of Computational Physics*, 258:613 – 633, 2014.
- ¹⁰K. Maute, M. Nikbay, and C. Farhat. Sensitivity analysis and design optimization of three-dimensional non-linear aeroelastic systems by the adjoint method. *International Journal for Numerical Methods in Engineering*, 56(6):911–933, 2003.
- ¹¹B. van Leer. Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov’s method. *Journal of Computational Physics*, 32(1):101 – 136, 1979.
- ¹²K. Wang, J. Grétarsson, A. Main, and C. Farhat. Computational algorithms for tracking dynamic fluidstructure interfaces in embedded boundary methods. *International Journal for Numerical Methods in Fluids*, 70(4):515–535, 2012.
- ¹³K. Wang, A. Rallu, J.F. Gerbeau, and C. Farhat. Algorithms for interface treatment and load computation in embedded boundary methods for fluid and fluidstructure interaction problems. *International Journal for Numerical Methods in Fluids*, 67(9):1175–1206, 2011.
- ¹⁴X. Zeng and C. Farhat. A systematic approach for constructing higher-order immersed boundary and ghost fluid methods for fluidstructure interaction problems. *Journal of Computational Physics*, 231(7):2892 – 2923, 2012.